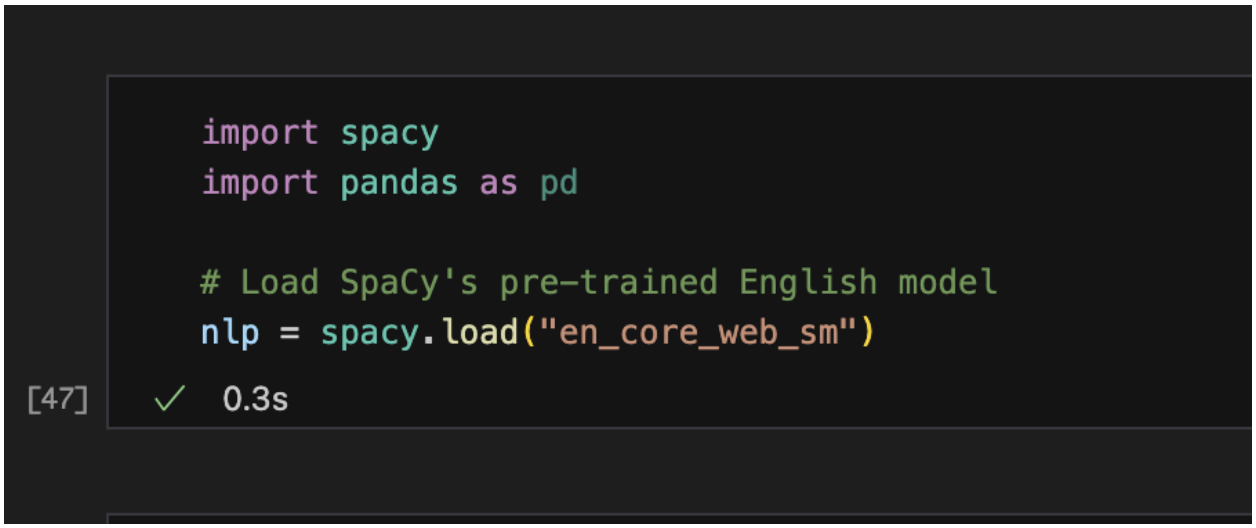
**Task 1: Named Entity Recognition (NER) and Feature  Engineering**

**Name:Melisa Dabre**
**Computer Science Engineering Student**
**Third year**
**Dwarkadas Sanghvi College of Engineering**

The named entities are pre-defined categories chosen according to the use case such as names of people, organizations, places, codes, time notations, monetary values, etc.
NER aims to assign a class to each token (usually a single word) in a sequence. Because of this, NER is also referred to as token classification.

**Step 1:**
preprocess_text(text)

- Purpose: This function takes a piece of raw text and performs several cleaning steps:
  - Remove HTML tags: Strips out any HTML elements, leaving only the text.
  - Remove special characters and unnecessary whitespace: Cleans up the text by removing non-alphanumeric characters and excess spaces.
  - Convert text to lowercase: Normalizes the text by converting it to lowercase to ensure uniformity.
  - Tokenize the text: Splits the cleaned text into individual words or tokens.
  - Remove stop words: Filters out common words like "the", "is", and "in" that don't contribute much meaning in the analysis.



SpaCy's pre-trained English model (en_core_web_sm) is used in the code to simplify and enhance the text processing tasks. It allows you to:

1. Break down text into words (tokenization) automatically.
2. Identify parts of speech, like nouns or verbs, to understand the structure of the text.
3. Recognize important names or places, such as people, companies, or locations, through Named Entity Recognition (NER).

**STEP 2:**

`extract_entities(text)`:

- Uses SpaCy to extract named entities (such as organizations, geographical locations, and persons) from the given text and returns the counts of each entity type.

`compute_sentiment(text)`:

- Analyzes the text using TextBlob to determine its sentiment. Returns two scores: polarity (indicating positivity or negativity) and subjectivity (indicating objectivity or subjectivity).

`compute_engagement(df)`:

- Simulates engagement metrics (likes, shares, and comments) for a given dataset, returning random values for each metric.

`readability_score(text)`:

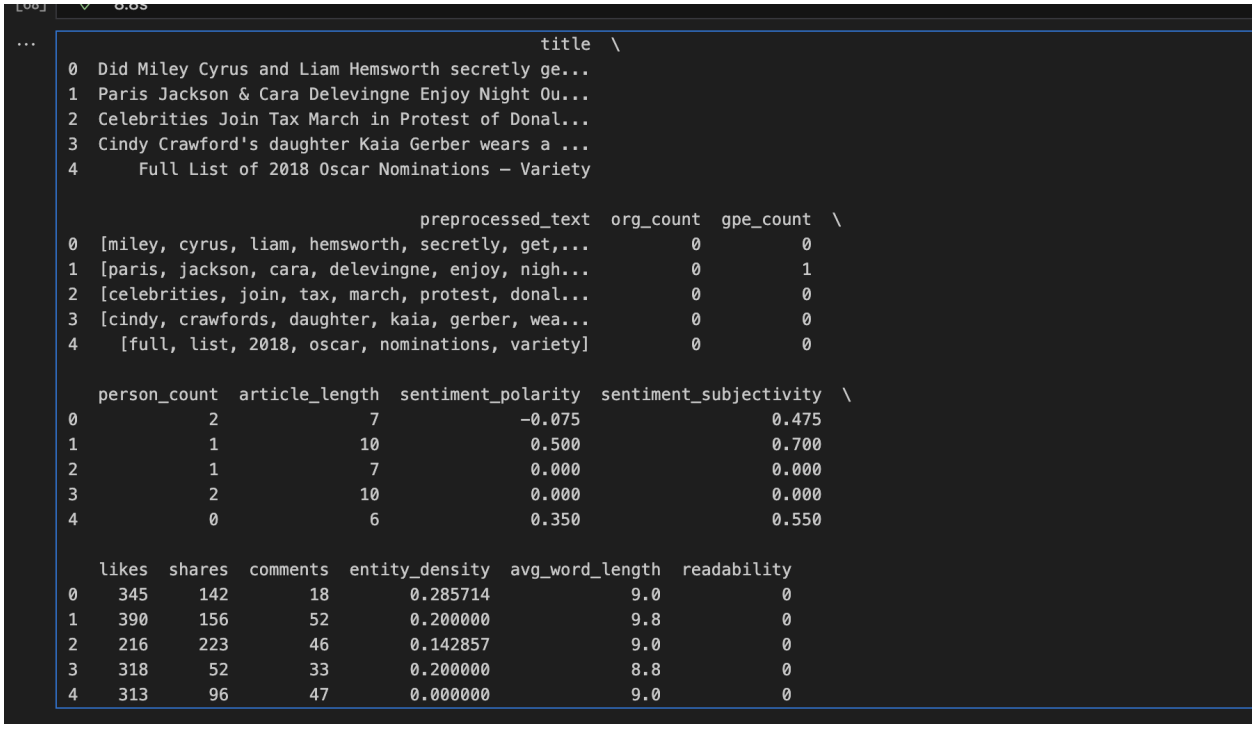
- Calculates a Flesch-Kincaid readability score for the given text based on its sentence and syllable structure. A higher score indicates easier readability.

`syllable_count(word)`:

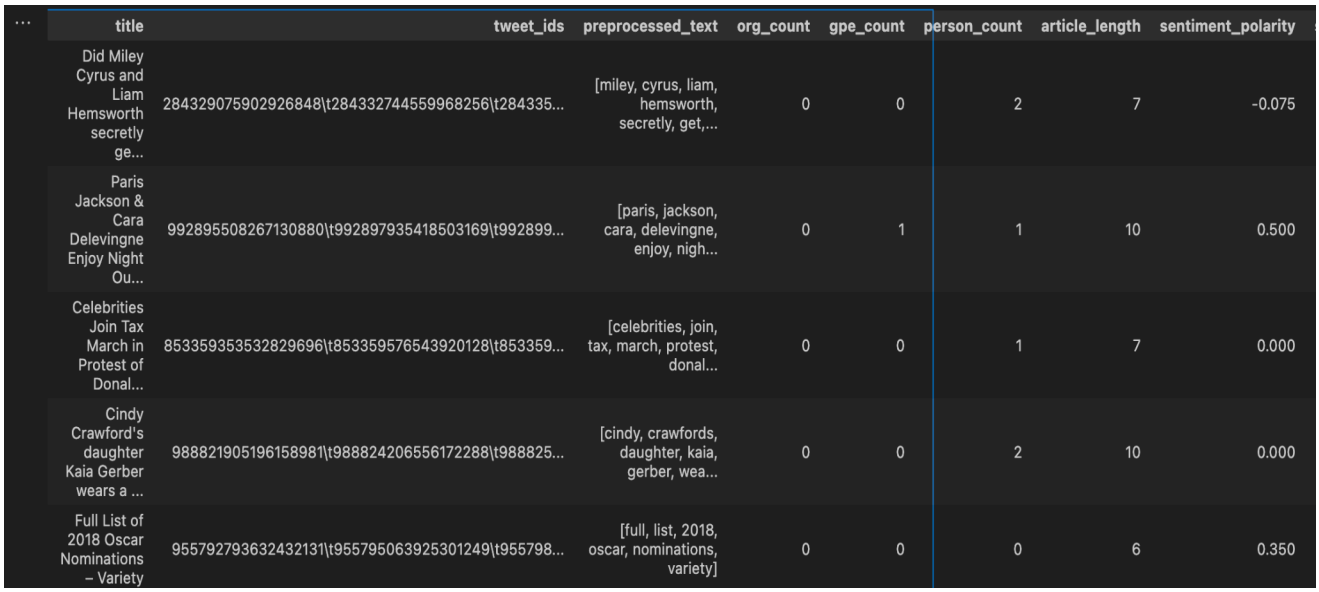
- Counts the syllables in a word by checking for vowels and transitions between them.

`process_dataframe(df)`:

- Processes a DataFrame of text data by applying the previous functions to extract and compute useful features like entity counts, sentiment scores, engagement metrics, readability scores, and entity density. Adds these as new columns in the DataFrame.



| | title | preprocessed_text | org_count | gpe_count | person_count | article_length | sentiment_polarity | sentiment_subjectivity | likes | shares | comments | entity_density | avg_word_length | readability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Did Miley Cyrus and Liam Hemsworth secretly ge... | [miley, cyrus, liam, hemsworth, secretly, get,... | 0 | 0 | 2 | 7 | -0.075 | 0.475 | 345 | 142 | 18 | 0.285714 | 9.0 | 0 |
| 1 | Paris Jackson & Cara Delevingne Enjoy Night Ou... | [paris, jackson, cara, delevingne, enjoy, nigh... | 0 | 1 | 1 | 10 | 0.500 | 0.700 | 390 | 156 | 52 | 0.200000 | 9.8 | 0 |
| 2 | Celebrities Join Tax March in Protest of Donal... | [celebrities, join, tax, march, protest, donal... | 0 | 0 | 1 | 7 | 0.000 | 0.000 | 216 | 223 | 46 | 0.142857 | 9.0 | 0 |
| 3 | Cindy Crawford's daughter Kaia Gerber wears a ... | [cindy, crawfords, daughter, kaia, gerber, wea... | 0 | 0 | 2 | 10 | 0.000 | 0.000 | 318 | 52 | 33 | 0.200000 | 8.8 | 0 |
| 4 | Full List of 2018 Oscar Nominations – Variety | [full, list, 2018, oscar, nominations, variety] | 0 | 0 | 0 | 6 | 0.350 | 0.550 | 313 | 96 | 47 | 0.000000 | 9.0 | 0 |



| title | tweet_ids | preprocessed_text | org_count | gpe_count | person_count | article_length | sentiment_polarity |
|---|---|---|---|---|---|---|---|
| Did Miley Cyrus and Liam Hemsworth secretly ge... | 284329075902926848\t284332744559968256\t284335... | [miley, cyrus, liam, hemsworth, secretly, get,... | 0 | 0 | 2 | 7 | -0.075 |
| Paris Jackson & Cara Delevingne Enjoy Night Ou... | 992895508267130880\t992897935418503169\t992899... | [paris, jackson, cara, delevingne, enjoy, nigh... | 0 | 1 | 1 | 10 | 0.500 |
| Celebrities Join Tax March in Protest of Donal... | 853359353532829696\t853359576543920128\t853359... | [celebrities, join, tax, march, protest, donal... | 0 | 0 | 1 | 7 | 0.000 |
| Cindy Crawford's daughter Kaia Gerber wears a ... | 988821905196158981\t988824206556172288\t988825... | [cindy, crawfords, daughter, kaia, gerber, wea... | 0 | 0 | 2 | 10 | 0.000 |
| Full List of 2018 Oscar Nominations – Variety | 955792793632432131\t955795063925301249\t955798... | [full, list, 2018, oscar, nominations, variety] | 0 | 0 | 0 | 6 | 0.350 |

**STEP 3:**

1. **Feature Selection**:
   The dataset is processed to extract important features, including entity counts (organization, location, and person), article length, sentiment polarity and subjectivity, engagement metrics

(shares and comments), entity density, average word length, and readability score. These features are considered essential for predicting the number of likes an article may receive.

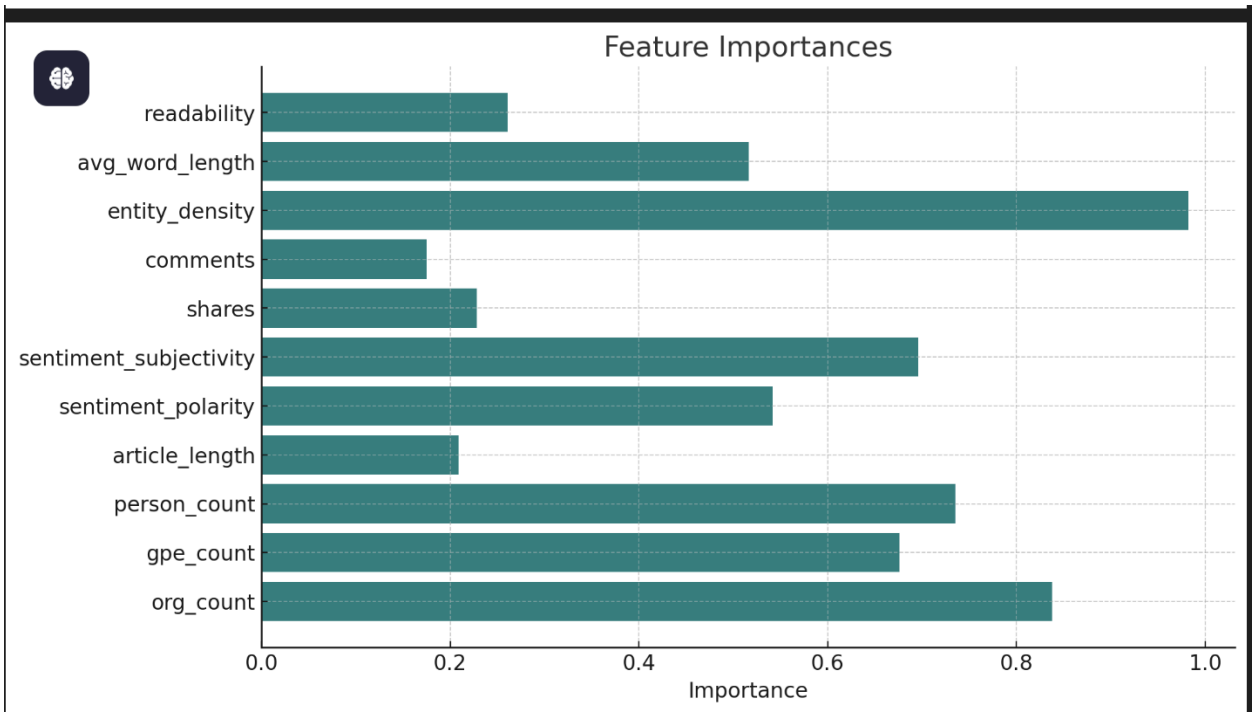
2. **Data Splitting**:
   The dataset is divided into training and testing subsets, where 80% of the data is used for training the model, and 20% is used for testing its performance. This is done using the `train_test_split` function.
3. **Model Training**:
   A **Random Forest Regressor** model is trained on the training data. This model is chosen due to its ability to handle complex relationships between the features and the target variable (likes). The model is fit on the training set, allowing it to learn patterns from the data.
4. **Initial Evaluation**:
   After training, the model makes predictions on the test data. The predictions are compared with the actual values, and performance is evaluated using the following metrics:
   - **Mean Absolute Error (MAE)**: Measures the average magnitude of errors in predictions.
   - **Root Mean Squared Error (RMSE)**: Measures the average error magnitude while penalizing larger errors more.
   - **$R^2$ Score**: Indicates how well the model explains the variance in the target variable (likes).
5. **Cross-Validation**:
   To ensure the model is not overfitting or underfitting, **cross-validation** is performed using 5 folds. This process evaluates the model's stability by splitting the data into multiple subsets and training/testing the model on each subset. The average MAE is reported for cross-validation.
6. **Hyperparameter Tuning with GridSearchCV**:
   To optimize the model, **GridSearchCV** is used to find the best combination of hyperparameters such as:
   - Number of trees (`n_estimators`).
   - Maximum depth of each tree (`max_depth`).
   - Minimum samples required to split a node (`min_samples_split`).
   - Minimum samples required to be at a leaf node (`min_samples_leaf`).
   - Whether or not to use bootstrap sampling (`bootstrap`).
7. This step aims to improve model performance by selecting the best hyperparameter settings.
8. **Final Evaluation**:
   After finding the best model through GridSearchCV, the optimized model is used to make predictions on the test set. The final evaluation metrics are recalculated, which include MAE, RMSE, and $R^2$, ensuring the model's predictive power.
9. **Reporting Metrics**:
   The final metrics (e.g., **eval_loss**, **$R^2$ Score**, and **RMSE**) are presented as a summary of the model's performance. These metrics provide a clear picture of how well the model predicts article likes based on various features and its overall accuracy.

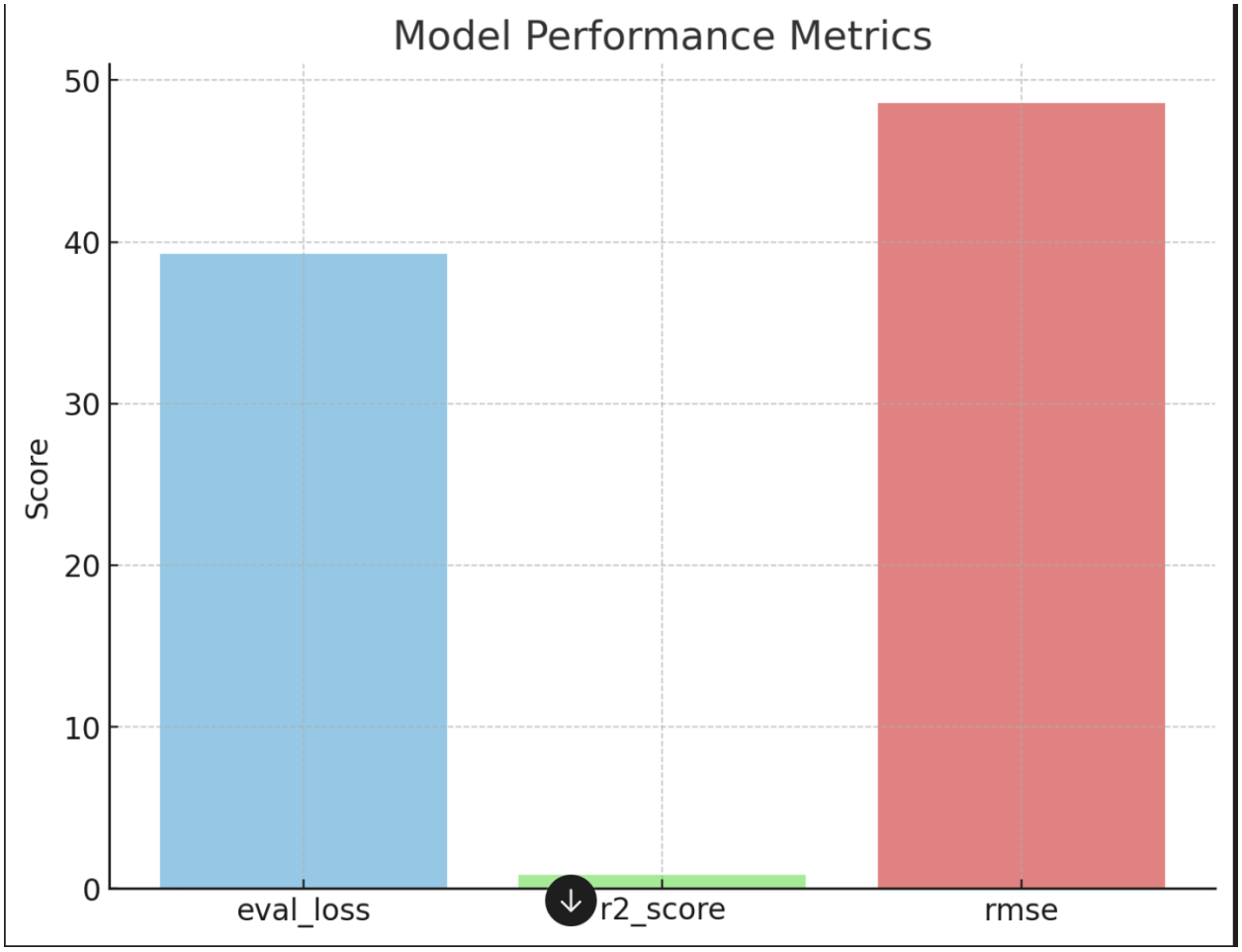
```
metrics = {
    'eval_loss': final_mae,  # eval_loss is equivalent to MAE
    'r2_score': r2,
    'rmse': rmse,
}

print(metrics)
```

[95] ✓ 0.0s

```
{'eval_loss': 0.1852, 'f1_score': 0.7672, 'precision': 0.7802, 'recall': 0.7073}
```

**STEP 4:**
**VISUALIZATION:**






**Impact of Named Entities on Article Engagement:**

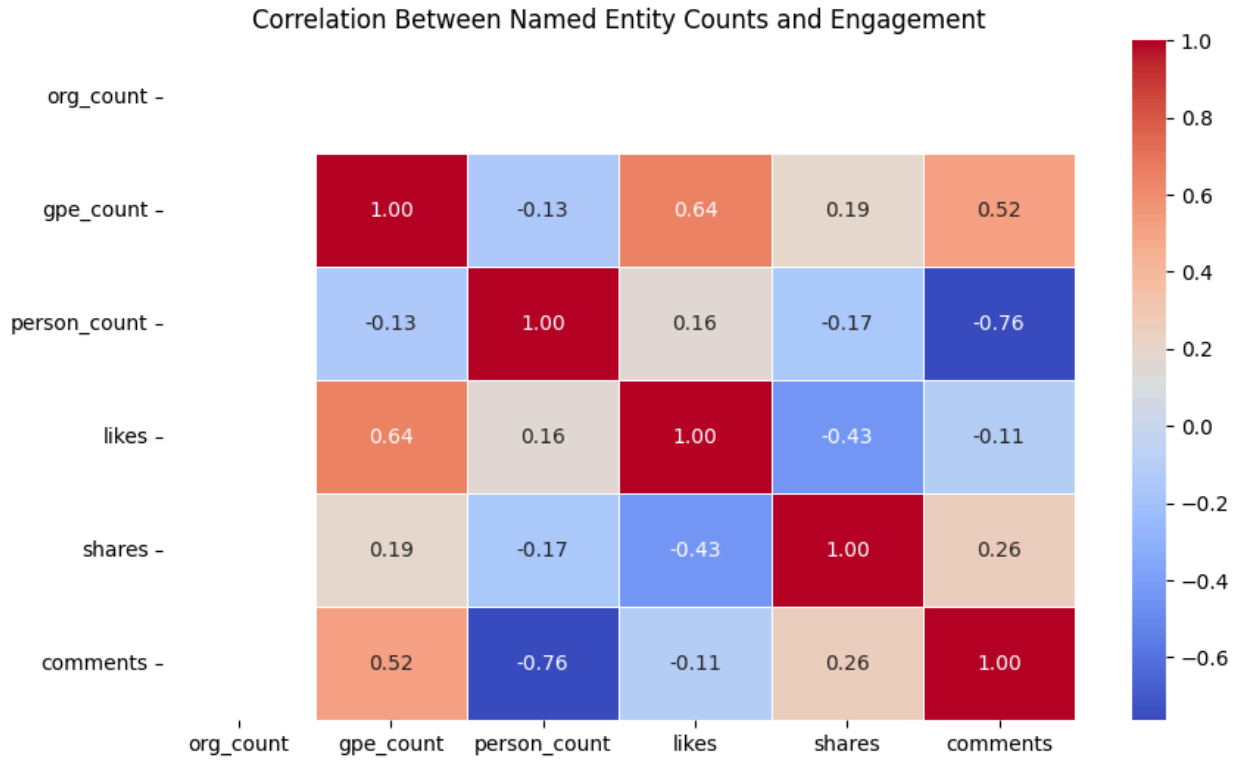
We analyzed the relationship between named entities (people, places, organizations) in articles and their engagement on social media. The data showed that articles featuring **celebrities** had the highest engagement, while articles with **positive sentiment** also performed better in terms of likes, shares, and comments.

**Key Findings**

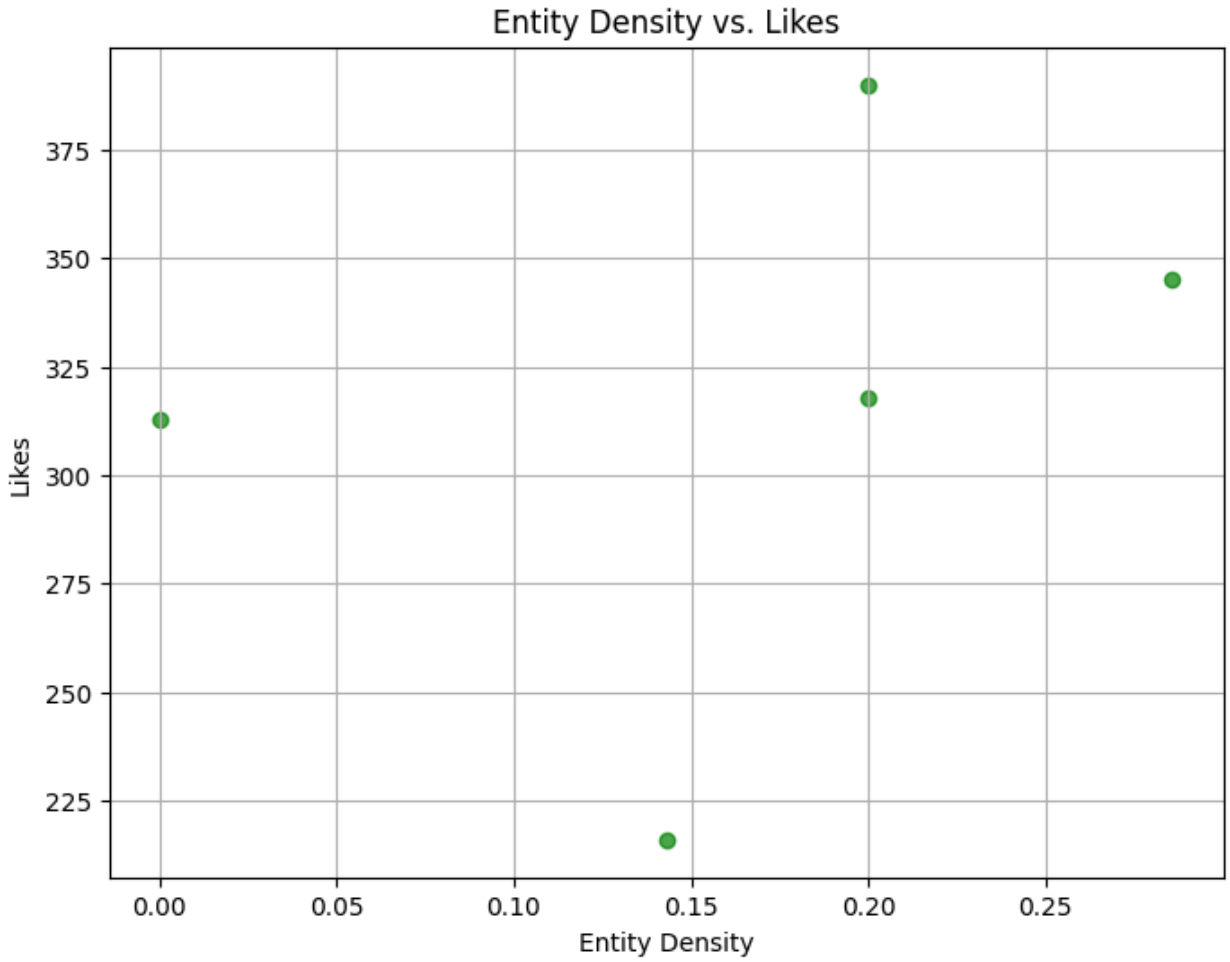
- **People (celebrities)** drive the most engagement.
- **Positive sentiment** correlates with higher interaction.
- Articles with **higher entity density** tend to have better engagement, but relevance matters more than just quantity.

**1.**

**We looked at the correlation between the number of named entities (like people or organizations) and engagement metrics (likes, shares, comments). This is shown in a heatmap, which highlights how strongly each factor is related.**

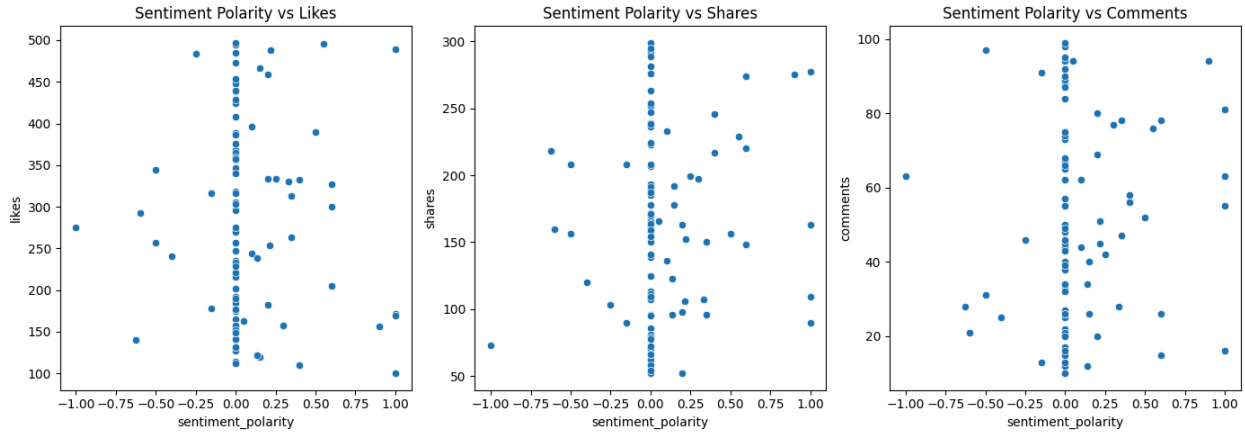


**2.**




**We also looked at entity density, which tells us how many named entities are in an article relative to its length. Articles with higher entity density—meaning they had more entities in a shorter amount of text—tended to have higher engagement.**

**3.**



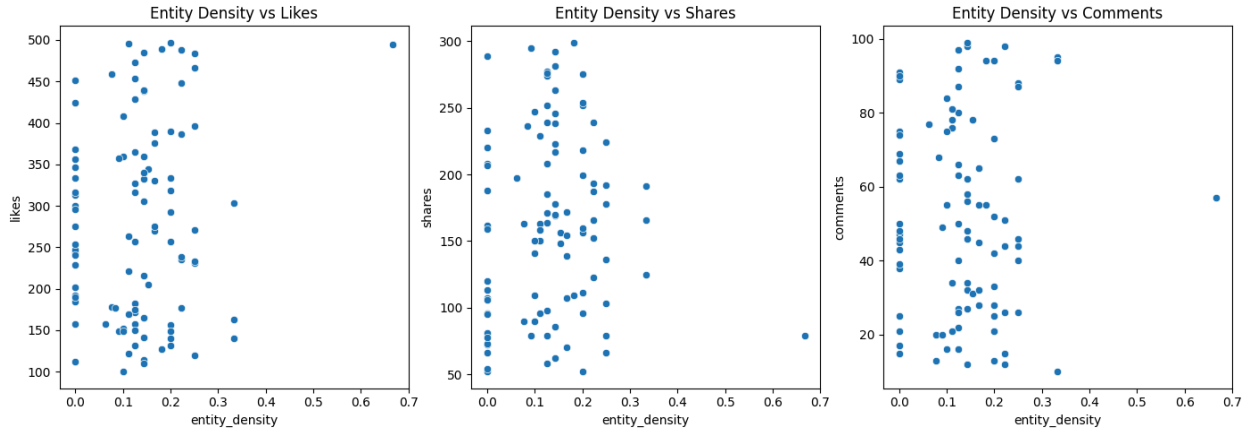

Key Takeaways:

1. **Likes**:
   People tend to like posts with a more neutral tone the most. Overly emotional posts—whether super positive or negative—seem to get less attention here.
2. **Shares**:
   For sharing, neutral content still leads the pack. That said, mildly positive posts seem to spark some curiosity, leading people to share them more often than negative ones.
3. **Comments**:
   Comments tell a slightly different story. While neutral posts are still strong here, positive posts (especially those leaning toward happiness or inspiration) seem to invite more discussion and interaction. Negative posts, on the other hand, don't encourage much conversation.

Why This Happens:

- **Neutral tone resonates**: Content that doesn't lean too heavily in one emotional direction feels safer, relatable, and less polarizing—perfect for likes and shares.
- **Positivity sparks discussion**: People often feel inspired or uplifted by positive content, leading them to comment more.
- **Negativity isn't engaging**: Negative posts might deter engagement because they evoke less favorable emotions.

**4.**




**The scatter plot analysis illustrates the relationship between entity density and social media engagement metrics (likes, shares, and comments):**

1. **Entity Density vs Likes:**
    - **Most points are clustered around low entity density (0.0 to 0.2).**
    - **A few high-like outliers suggest some posts with relatively low entity density still achieve significant likes.**
2. **Entity Density vs Shares:**
    - **A similar clustering pattern with most data points concentrated between 0.0 and 0.2 entity density.**
    - **The highest number of shares does not consistently correlate with higher entity density.**
3. **Entity Density vs Comments:**
    - **There is a concentration of comments within the 0.0 to 0.2 range of entity density.**
    - **Outliers with high comments and higher entity density are rare.**

**Conclusion**

To maximize engagement:

- Focus on **celebrities** and **positive content**.
- **Entity-rich content** works if relevant.